

多価関数を近似するフィードバック付き階層形ニューラルネットワーク

著者	富川 義弘, 中山 謙二
雑誌名	電子情報通信学会論文誌. D-II, 情報・システム, II-情報処理
巻	J80-D-2
号	9
ページ	2493-2501
発行年	1997-09-25
URL	http://hdl.handle.net/2297/5616

多価関数を近似するフィードバック付き階層形ニューラルネットワーク

富川 義弘^{†*}

中山 謙二^{††}

A Multilayer Neural Network with a Feedback Loop Approximating Many-Valued Functions

Yoshihiro TOMIKAWA^{†*} and Kenji NAKAYAMA^{††}

あらまし 階層形ニューラルネットワーク (NN) は、任意の 1 価関数を近似することができる。しかし、多価関数近似に拡張する場合には、特殊な構造、および、構造に対応した学習法が必要になる。本論文では、出力層から入力へのフィードバック構造を取り入れた階層形 NN で、多価関数を近似する方法を提案する。出力層から入力へのフィードバックによるダイナミクスは、数値解法の逐次代入処理に相当する。提案モデルは、多価関数の近似曲線を、学習されたネットワーク関数の非線形方程式の解として検出する。ネットワークの学習には、補助入力をもつ階層形モデルを使用し、出力層への教師データを補助入力に与える。また、関数近似曲線への引込み動作を学習するために、学習時に漸近条件を付加する。漸近条件は、次の二つの学習方式によって与えられる。(1) データ点近傍で、補助入力に対する偏微分が 1 以下になるように、荷重更新則に偏微分条件を付加する。(2) 補助入力に加える学習データをわずかに変動させる。補助入力を導入することによって、多価関数は 1 価関数に置き換えられ、階層形 NN での学習が可能になる。その結果、データのクラスタリングやセグメンテーション等の前処理を必要としないばかりでなく、同一構造のネットワークで、任意の価数の多価関数を近似できる。提案モデルによる多価関数近似は、計算機シミュレーションで確認された。

キーワード 階層形ニューラルネットワーク、フィードバック、多価関数、関数近似、非線形方程式、逐次代入法

1. ま え が き

関数近似とは、与えられたデータ点の間を補間する関数を求める処理である。単調増加の非線形連続関数の線形和で任意の 1 価連続関数を近似できることは、ユニバーサル近似理論として知られている [1]。ニューロンの反応は、一種の非線形関数であり、階層形ニューラルネットワーク (NN) の出力を、隠れ層ニューロン出力の線形和とすれば、階層形 NN を関数近似モデルとして利用することができる。

階層形 NN による関数近似のメリットは、その学習能力により、任意の関数を近似できることである。これは、予測問題や類推問題に応用することができる。

本論文で対象とする多価関数は、一つの独立変数の値に対し、複数の値をとる関数である。従来の階層形 NN では、同じ入力 (独立変数) に対して異なる出力 (関数値) を得ることは、原理的に不可能である。従って、多価関数を近似するためには、ネットワークに特殊な構造、および、構造に応じた学習方法が必要になる。多価関数近似に適用するために、従来から、いくつかのモデルが提案されている。

複数の出力ユニットをもつ階層形 NN で多価関数値を近似する方法は、最も素直な方法である。これは、出力をベクトル \mathbf{y} とし、入力値 x に対し、 $\mathbf{y} = f(x)$ の関数で近似することに相当する。しかし、関数近似のデータがランダムに与えられる場合、この方法では、あらかじめデータを出力ユニットごとに分類 (クラスタリング) しておく必要がある。志沢 [5] は、クラスタリング処理が難しい問題であることを指摘している。

また、Jacobs ら [2] は、近似領域を分割 (セグメンテーション) し、モジュールを配置して多価問題に対処する方法を提案している。この方法も、セグメン

[†] 金沢大学大学院自然科学研究科, 金沢市
Kanazawa University, 2-40-20 Kodatsuno, Kanazawa-shi, 920 Japan

^{††} 金沢大学工学部電気・情報工学科, 金沢市
Kanazawa University, 2-40-20 Kodatsuno, Kanazawa-shi, 920 Japan

* 現在, YKK 株式会社研究開発本部

テーション処理に、クラスタリング処理と同等の難しさがある。

多価関数を時系列データと考え、時間遅延ネットワーク (TDNN) を利用することもできる。Linら [3] は、TDNN を改良した適応 TDNN (ATNN) を用いて、多価関数をトレースする方法を提案している。この方法は、データをクラスタリングする必要はないが、データの順序情報を把握しておく必要がある。

これらの多価関数近似モデルは、データの分類や順序に関する情報が、あらかじめわかっている場合（先見情報がある場合）には有効である。しかし、実際の応用では、データはランダムに与えられ、データ間の関係を把握することが困難な場合がある。本論文では、この種の問題を考える。

志沢 [4], [5] は、RBF ネットワークと高次多項式の数値解法を利用して、多価関数を近似する方法を提案した。志沢のモデルは上記のような先見情報を必要としないという意味で有効なモデルである。しかし、高次多項式の数値解法を前提とした手法であるため、価数がネットワークで限定されるという問題点がある。

本論文では、出力層から入力へのフィードバック構造を取り入れた階層形 NN で、多価関数を近似する方法を提案する。フィードバック機構によるダイナミクスは、逐次代入法による非線形数値解法と等価である。提案モデルは、学習された出力関数の非線形方程式の解として、多価関数を近似する。志沢の多項式モデルとは異なり、近似関数に関する制約がないので、任意価数の関数を同一構造のネットワークで近似することができる。

本論文では、まず、2. で、提案モデルと学習方式を述べる。3. で、志沢のモデルとの原理的違いを比較する。4. で、シミュレーションによって、提案方式により、価数変化のない任意の多価関数を、同一構造のモデルで近似できることを示す。5. で、提案モデルの問題点と今後の可能性について考察する。

2. フィードバック付き階層形 NN

2.1 補助入力学習モデルと基本学習方式

提案モデルの関数近似手法は、階層形 NN の学習法を多価関数に発展させたものである。まず、多価性をもつ図 1(a) のデータを、図 1(b) の単一入出力の階層形 NN で学習する場合の問題点を考える。データは、独立変数 x^* に対する関数値 y^* のセットである。本論文では、関数近似のデータを、*付きで示す。

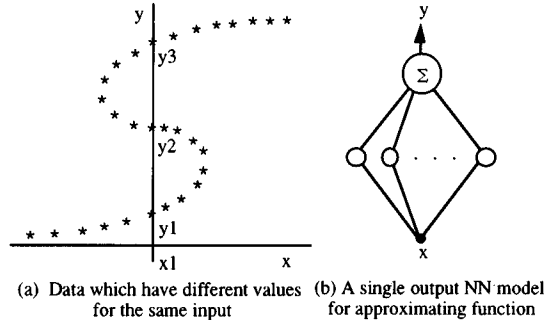


図 1 多価関数近似データと単一入出力階層形モデル
Fig. 1 Data and a single output model for approximating functions.

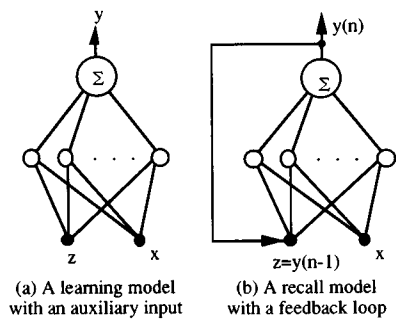


図 2 補助入力学習モデルと想起モデル
Fig. 2 A learning model with an auxiliary input and a recall model.

図 1(a) のようなデータを学習する場合、同じ入力 $x1^*$ に対して、複数の異なる関数値 $y1^*, y2^*, y3^*$ が、教師データとして与えられることになる。単一入出力の階層形 NN が学習できるのは、1 対 1、または、多対 1 の関係のみである。従って、単一入出力の階層形 NN で、多価性のあるデータを近似することは原理的に不可能である。

この問題を解決するために、図 2(a) のように、階層形 NN の入力に補助入力 z をもつモデルを考える。この補助入力モデルの出力関数は、次式で表すことができる。

$$y = f(x, z) \quad (1)$$

このモデルで多価関数を近似する場合、入力ベクトル (x, z) 対出力 y の関係が 1 対 1 対応になるように、補助入力 z を決定できれば、原理的に学習することができる。この補助入力モデルの問題は、補助入力 z の決定方法である。

提案モデルは、補助入力 z として、出力教師データ

と同じ y^* を用いて学習する。つまり、

$$\text{入力ベクトル: } (x^*, y^*) \rightarrow \text{出力: } y^* \quad (2)$$

与えられる対応関係を補助入力モデルで学習する。この学習で、補助入力の y^* と出力教師データの y^* は、常に 1 対 1 対応しているので、入力ベクトル (x^*, y^*) と教師データ y^* は、入力 x^* に無関係に 1 対 1 対応する。従って、この学習方式を利用すれば、多価性をもつデータをランダムに与えても、補助入力モデルに学習させることができる。

2.2 逐次代入法と想起モデル

問題は、式 (2) の対応関係を学習した補助入力モデルから、任意の独立変数 x が与えられたときに、どのように関数近似値 y を想起するかにある。

学習が終了したとき、式 (1) の補助入力モデルは、与えられたすべてのデータ (x^*, y^*) に対して、次式の関係を満たしている。

$$y^* = f(x^*, y^*) \quad (3)$$

そこで、このように学習された式 (1) の関数に対して、補助変数 z を変数 y に置き換えた非線形方程式、

$$y = f(x, y) \quad (4)$$

の解 y を任意の x に対して求めることができれば、この解 y は、少なくとも学習データ x^* に対しては y^* であり、学習されていない x に対しては、求める関数の近似値であることが期待される。

提案モデルは、この非線形方程式を解くために、次式の逐次代入法 [7] を導入する。

$$y(n) = f(x, y(n-1)), n \rightarrow \infty \quad (5)$$

この処理は、図 2(b) に示すようなフィードバック機構をもつ階層形 NN に置き換えることができる。提案モデルは、ダイナミックスをもつという意味では、リカレント形モデルの性質をもっている。しかし、隠れ層ニューロンの総和として出力を決定するという意味では、階層形モデルである。従来モデルと明確に区別するために、フィードバック付き階層形 NN と呼ぶことにする。

式 (5) の求解法で、解をデータ点に収束させるためには、学習した階層形 NN の出力関数 $f(x, z)$ が、データ点 (x^*, y^*) の近傍で、次式の条件を満たしている必要がある [7]。

$$|f_z(x^*, y^*)| < 1 \quad (6)$$

ここで、 $f_z(x, z)$ は、関数 $f(x, z)$ の補助変数 z に関する偏微分とする。以降、 z に関する偏微分に関しては、この表記を利用する。

$$f_z(x, z) = \frac{\partial f(x, z)}{\partial z} \quad (7)$$

式 (6) の条件を、本論文では、漸近条件と呼ぶことにする。提案モデルは、この漸近条件を学習条件に含め、学習点への収束を確保する。この漸近条件の導入法に関しては、次節で述べる。

2.3 漸近条件の導入

フィードバック付き階層形 NN で、出力を関数近似点に収束させるためには、2.1 で述べた学習法に加えて、式 (1) の関数形が、式 (6) の漸近条件を満たしている必要がある。

式 (6) は不等号条件であるため、そのままネットワークに与えることは難しい。そこで、次式の等号条件を目標に学習するものとする。

$$f_z(x^*, y^*) = 0 \quad (8)$$

ネットワークに式 (8) の条件を与えるために、荷重の更新則に偏微分条件を導入する方法と変動データを付加して学習する方法を提案する。シミュレーションからも明らかになるが、これら二つの方法を併用することで、近似曲線への収束性が向上する。

2.3.1 荷重更新則に偏微分条件を導入する方法

荷重更新則について述べるために、図 2(a) の補助入力モデルを想定し、入力データ (x_j^*, y_j^*) に教師データ y_j^* を学習するものとする。学習データ数は M とする。補助入力モデルの隠れ層のニューロン数を N とし、それぞれの活性化関数を ϕ_i で表記する。隠れ層のニューロンから出力層への結合荷重を a_i 、出力層のしきい値を b としておく。また、隠れ層の i 番目のニューロンにおける入力 x に対する結合荷重を w_{ix} とし、補助入力 z に関する荷重を w_{iz} とし、しきい値を θ_i で表記する。

i 番目の隠れ層ニューロンの出力は、 $\phi_i(w_{ix}x + w_{iz}z + \theta_i)$ で表せるので、入力 (x, z) に対する出力関数 $f(x, z)$ は、次式のようにになる。

$$f(x, z) = \sum_{i=1}^N a_i \phi_i(w_{ix}x + w_{iz}z + \theta_i) + b \quad (9)$$

また、出力関数 $f(x, z)$ の補助変数 z に関する偏微分

$f_z(x, z)$ は、次式で与えられる。

$$f_z(x, z) = \sum_{i=1}^N a_i w_{iz} \psi_i(w_{ix}x + w_{iz}z + \theta_i) \quad (10)$$

ここで、 ψ_i は、 ϕ_i の微分で、 $\psi_i(x) = \frac{d\phi_i(x)}{dx}$ である。

学習の目的は、出力値を学習データに一致させ、かつ、式 (8) の偏微分として与えられる等号条件にできるだけ近くなるような荷重 a_i , w_{ix} , w_{iz} としきい値 b , θ_i を求めることである。そこで、出力誤差関数 E_e および偏微分誤差関数 E_d を次式で定義しておく。

$$E_e = \sum_{j=1}^M (y_j^* - f(x_j^*, y_j^*))^2 \quad (11)$$

$$E_d = \sum_{j=1}^M (f_z(x_j^*, y_j^*))^2 \quad (12)$$

上記の目的の十分条件は、出力誤差関数 E_e と偏微分誤差関数 E_d が、同時に 0 になることである。そこで、次式の統合誤差関数 E を定義し、この統合誤差関数 E が、最小になるように荷重としきい値を決定する。

$$E = E_e + \beta E_d \quad (13)$$

ここで、 β は 1 以下の正定数である。漸近条件は、式 (6) の不等号条件であるため、必ずしも式 (8) の等号条件を満たす必要はない。そこで、 E_d 項の重みを小さくし、 E_e 項を重視した学習を行うようにする。

統合誤差関数 E の最小化には、最急降下法を利用する。つまり、次式で決定される微小変量を用いて、荷重としきい値を少しずつ更新する。

$$\Delta a_i = -\eta \frac{\partial E}{\partial a_i} = -\eta \left(\frac{\partial E_e}{\partial a_i} + \beta \frac{\partial E_d}{\partial a_i} \right) \quad (14)$$

$$\Delta b = -\eta \frac{\partial E}{\partial b} = -\eta \left(\frac{\partial E_e}{\partial b} + \beta \frac{\partial E_d}{\partial b} \right) \quad (15)$$

$$\Delta w_{ix} = -\eta \frac{\partial E}{\partial w_{ix}} = -\eta \left(\frac{\partial E_e}{\partial w_{ix}} + \beta \frac{\partial E_d}{\partial w_{ix}} \right) \quad (16)$$

$$\Delta w_{iz} = -\eta \frac{\partial E}{\partial w_{iz}} = -\eta \left(\frac{\partial E_e}{\partial w_{iz}} + \beta \frac{\partial E_d}{\partial w_{iz}} \right) \quad (17)$$

$$\Delta \theta_i = -\eta \frac{\partial E}{\partial \theta_i} = -\eta \left(\frac{\partial E_e}{\partial \theta_i} + \beta \frac{\partial E_d}{\partial \theta_i} \right) \quad (18)$$

ここで、 η は、学習係数であり、小さな正定数である。式 (14)~(18) で与えられる修正量は、通常の誤差逆伝搬学習法 (BP 法) の修正量に漸近条件に関する偏微分誤差修正項が付加された形になる。

2.3.2 変動データを用いる方法

式 (8) を学習データ点 (x^*, y^*) の近傍で差分近似すると、次式のようになる。

$$\begin{aligned} f_z(x^*, y^*) \\ \simeq \frac{f(x^*, y^* + \Delta z) - f(x^*, y^*)}{\Delta z} = 0 \\ f(x^*, y^* + \Delta z) = f(x^*, y^*) = y^* \end{aligned} \quad (19)$$

つまり、学習時の入力 (x^*, y^*) に加えて、補助入力 z に関して、 y^* をわずかに変化させた入力を与え、同じ教師データ y^* を与えることで、近似的に式 (8) の等号偏微分条件をネットワークに与えることもできる。

この方法は、2.3.1 の方法に比較して、漸近条件を与える効果は小さい。しかし、データを補助入力 z に関して分散させることで、引込み領域を広げる効果がある。従って、両学習手法を併用することで、安定かつ広い引込み領域を形成することができる。

2.4 ネットワークの学習とダイナミクス

図 3(a) は、2.1, 2.3 で述べた手法を用いた学習により得られる関数形 $y = f(x, z)$ を示している。図中“*”で示される点々が、データに相当する。多価性のあるデータの場合、同一の x^* に対して、複数の y^* が存在する。しかし、補助入力 z を加え、入力の次元を増やすことにより、変数ベクトル (x, z) に対する 1 価関数となる。階層形 NN で、任意の 1 価関数を近似できることは理論的にも証明されており [1]、提案学習方式で多価関数を補助入力モデルに学習させることは可能である。また、漸近条件を導入することにより、関数形は階段状になる。

図 3(b) は、2.2 で述べたフィードバック付き階層形 NN の想起時のダイナミクスを示したものである。

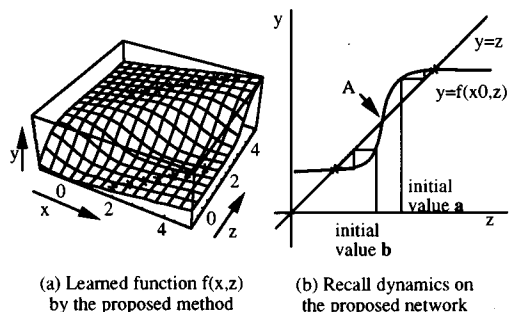


図3 学習されたネットワーク関数とダイナミクス
Fig.3 Learned network function and dynamics by the proposed network.

想起の場合、出力 y を補助入力 z にフィードバックする。この処理は、補助線 $y = z$ で出力 y を入力 z に折り返す処理になる。また、この処理において、入力 x は固定して考えることができるので、図 3(b) のような 2 次元断面上でダイナミックスを議論できる。

データは、補助入力と出力が同じになるように与えられる。従って、近似点は出力関数と補助線 $y = z$ の交点になる。この例のように、近似点近傍で平坦な関数形が形成されている場合、分岐点 A を境に、初期値によって異なるデータ点に収束させることができる。提案モデルは、この状態分岐を利用して、多価関数を近似する。

3. 従来手法との比較

3.1 志沢のモデルとの比較

ニューラルネットを用いた多価関数近似に関して、従来モデルと提案モデルを比較する。従来の多価関数の近似モデルとして、1. でも述べたように、Jacobs ら [2] のモジュラーネットワークや Lin ら [3] の ATNN 等が挙げられるが、これらのモデルは、データに関しての先見情報が必要であるという点で対象とする領域が異なる。ここでは、志沢 [4], [5] のモデルを取り上げ、提案モデルとの違いを考察する。

志沢のモデルは、入力 x と出力 y との関係が次式の N 次多項式を用いて、陰関数 $F(x, y) = 0$ として与えられることを前提にしている。

$$\begin{aligned} F(x, y) &= \prod_{i=0}^N (y - f_i(x)) \\ &= y^N + \sum_{i=0}^{N-1} (-1)^{i-1} g_i(x) y^{i-1} = 0 \end{aligned} \quad (20)$$

ここで、 $f_i(x)$ は、 x が定まったときの多価近似値の一つを表し、 $g_i(x)$ は、 $F(x, y)$ を y に関する N 次多項式で表現したときの係数である。

式 (20) において、係数関数 $g_i(x)$ を任意の x に対して求めることができれば、高次多項式の数値解法を利用して、変数 y の値を決定することができる。志沢は、一般正則化理論を拡張して、任意の入力 x から係数関数 $g_i(x)$ を出力する RBF ネットワークを構成し、多価性のあるデータからそのネットワークを学習する方法を提案した。

つまり、係数関数 $g_i(x)$ を、点 c_j を中心とする放射

基底関数 $R(x, c_j)$ の線形和として、次式のように展開し、関数 $g_i(x)$ を算出するネットワークを構成する。

$$g_i(x) = \sum_{j=0}^{M-1} a_{ij} R(x, c_j) \quad (21)$$

ここで、 a_{ij} はネットワーク固有の定数である。与えられたデータから、この定数が決定できれば、任意の x に対する係数関数 $g_i(x)$ は、式 (21) より求めることができる。

式 (21) を式 (20) に代入すれば、次式のように、陰関数 $F(x, y)$ は、係数 a_{ij} に関して 1 次関数となる。

$$\begin{aligned} F(x, y) \\ = y^N + \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} a_{ij} R(x, c_j) (-1)^{i-1} y^{i-1} = 0 \end{aligned} \quad (22)$$

係数 a_{ij} は、陰関数 $F(x, y)$ をできるだけ 0 に近づけるように決定する。つまり、次式の誤差関数 E を最小化するように決定すればよい。

$$E = \sum \{F(x^*, y^*)\}^2 \quad (23)$$

陰関数 $F(x, y)$ は、係数 a_{ij} に関して 1 次関数であるので、簡単な線形処理で最小化できる。多価関数を陰関数として近似する方法は、円近似などで利用される。志沢のモデルは、これを一般的な多価関数近似に拡張したものと考えることができる。

志沢のモデルの長所であり短所でもある点は、陰関数 $F(x, y)$ を、式 (20) の高次多項式に規定している点である。この仮定により、学習を簡単化できる。また、過去の高次多項式の数値解法が利用できるため、複素解を求めたり、複数の多価出力を同時に求めることも可能になる。しかし、多価関数を高次多項式に当てはめるために、学習できる価数が仮定した多項式の次数で限定される。つまり、データの価数に関しては、情報として与えられなければならない。

これに対して、提案モデルは、式 (4) に示すように、

$$y = f(x, y)$$

の解として、多価関数を近似する。この関数 $f(x, y)$ は、階層形 NN の出力であり、データに応じて適応的に決定できる。つまり、階層形 NN の柔軟性を、そのまま、多価関数の近似に利用することができる。従っ

て、同一構造のネットワークで、任意の価数の多価関数を学習することができる。反面、高次多項式の解法が利用できないため、複数の多価関数値を同時に得るようなことはできない。

3.2 逐次代入法とニュートン法

式 (4) の非線形方程式は、次式のニュートン法を利用して、解くことができる。

$$y(n) = y(n-1) - \frac{y(n-1) - f(x, y(n-1))}{1 - f_z(x, y(n-1))}, \quad n \rightarrow \infty \quad (24)$$

ニュートン法を利用した場合は、データ点への引込み条件は、解析的に決定されるので、2.3の漸近条件を学習する必要はない。しかし、式 (24) の関数には特異性があるため、この処理をネットワーク化することは不可能である。また、ニュートン法は収束速度は速いが、解への収束が保証されないことは知られている。

これに対し、提案モデルは、逐次代入法を導入し、想起モデルもネットワーク化した。また、学習に漸近条件を導入し、引込み条件も学習に含めた。収束特性は、すべて、学習で決定される。

提案モデルのようなダイナミックスをもつモデルは、シミュレーションでも示すように、不安定動作を示すことがある。これは、ニュートン法でも同じである。

このようなモデルにおいて、不安定動作、引込み領域の制御は重要な問題である。提案モデルは、学習条件により引込み領域を制御することができる。この点で、ニュートン法よりも、不安定動作を回避できる可能性は高い。

4. シミュレーション

4.1 多価関数の学習と引込み領域

提案モデルによる多価関数の近似をシミュレーションで示す。隠れ層のニューロンは、15個用意し、活性化関数にはシグモイド関数を使用した。式 (13) の偏微分誤差関数の重み β には、0.2を用いた。

図4は、近似曲線図に“*”で示される直線上に並んだデータを用いて、関数近似を行ったときの結果である。学習は、2.3で述べた、(a) 変動データを用いる方法、(b) 更新則に偏微分条件を導入する方法と (c) その両者を併用した3通りの方法で行った。図4の近似曲線は、初期値 (x, y) を、 xy 空間上に一様に発生させ、ネットワークを動作させることによって得られる出力 y の収束点を“.”でプロットしたものである。

図4(a)の場合、学習曲線は最初急激に低下するが、その後、誤差値は不変で10万回の更新によっても収束しなかった。近似曲線も、データと一致する部分はほとんどなかった。

これに対し、偏微分条件を導入した図4(b), (c)の

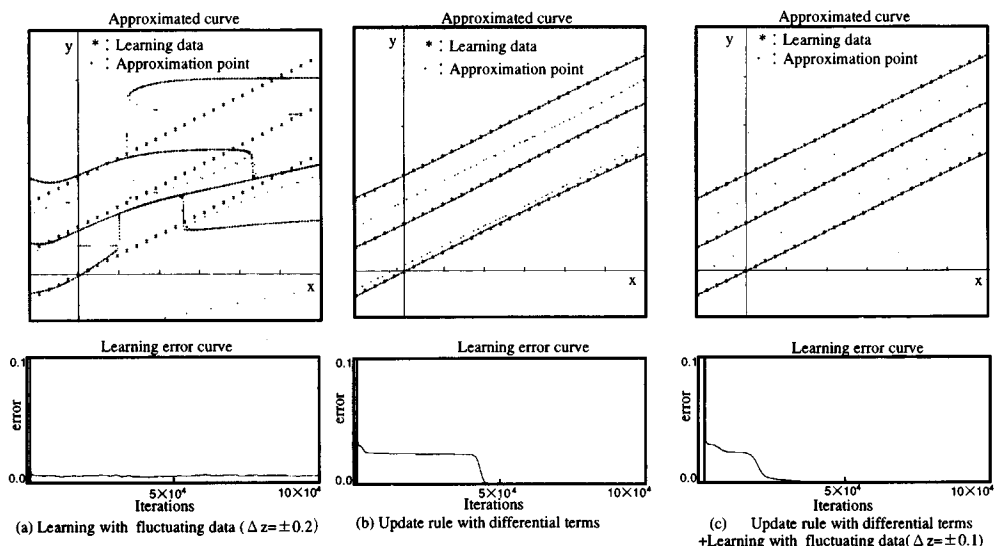


図4 学習法による学習曲線と近似結果の比較
Fig.4 Learning curves and approximated results by 3 learning methods.

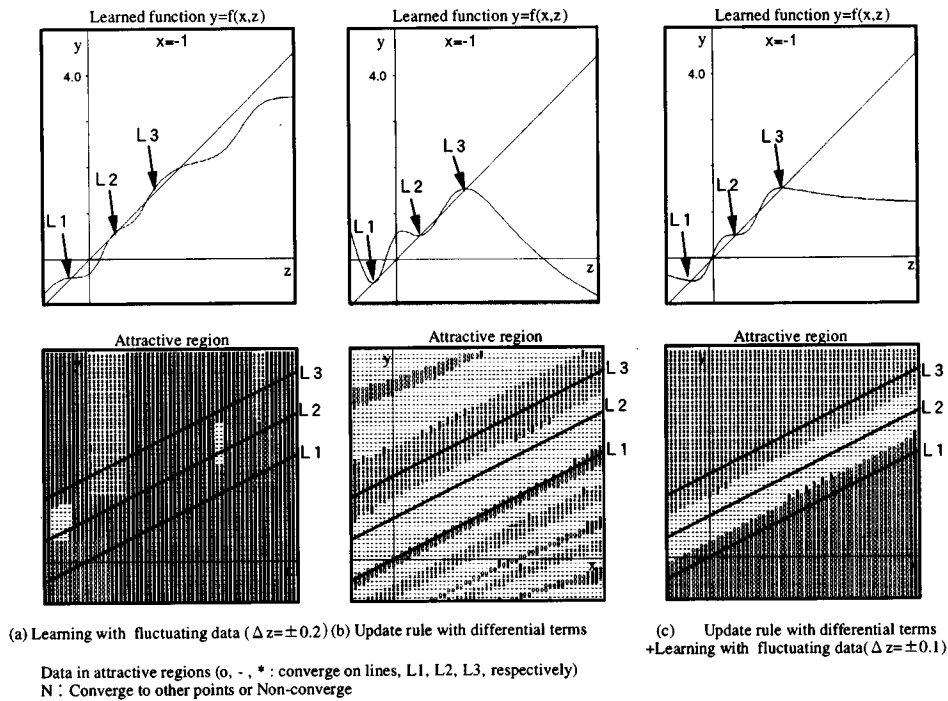


図5 学習法による関数形と引込み領域の比較
Fig.5 Learned function shapes and attractive regions by 3 learning methods.

場合、学習曲線は、最初、図 4(a) に比較してかなり高い誤差値で、いったん停留するが、約 2.5~4 万回を過ぎたあたりで再び低下し始め、最終的に十分な収束が得られた。近似曲線も、三つの直線が正しく近似できた。

図 4(a) の学習で収束が得られなかった理由は、変動データが与える漸近条件に関する修正効果が小さすぎるためである。提案モデルは、まず、出力値をデータに一致させるように変化し、その後、漸近条件を満たすように状態を更新していく。これは、誤差曲線の階段状の変化となって現れる。図 4(a) の場合、漸近条件に関する誤差量は、図 4(b), (c) に比較して、かなり小さい。この結果、漸近条件の学習の進行が遅れ、収束できなかったものと判断している。

図 5 は、3 通りの学習方法によって得られたネットワークの、初期値と引込み領域の関係、および、学習された関数形を $y = f(-1, z)$ のグラフから比較したものである。

図 5(b) の場合、L1, L2, L3 の 3 直線への引込み領域が、しま状に分布している。これは、引込み領域が飛び飛びになっていることを示している。 $y = f(-1, z)$

の関数形から推察すると、L1, L3 の直線に対応する点では、偏微分条件は満たされてはいるが、ちょうど、凸と凹の部分に一致している。つまり、偏微分条件を満たす部分が、学習点近傍のみに限定されていることになる。多くの応用において、初期値に最も近い解が求められる。学習データ点だけに偏微分条件を与えるという方法では、条件を満たす領域をデータ点近傍に限定してしまう。これに対し、両方の学習方式を取り入れた図 5(c) では、引込み領域の境も 3 本の直線の中央に分布しており、望ましい学習結果である。この場合、変動データは、引込み領域を広げるために利用されたことになる。

図 4 の近似曲線の図で、近似曲線の中にノイズのような収束点があるが、これは図 5 の引込み領域が分岐するあたりに相当する。分岐点では、状態変化の移動が小さく収束点とみなされてしまう場合がある。

4.2 任意価数の多価関数の近似

提案モデルが、任意の価数、任意形状の多価関数近似の問題に適用できることを示すために、この節では、前節以外の多価データに対するシミュレーション結果を示す。この節のシミュレーションは、前節と同じネット

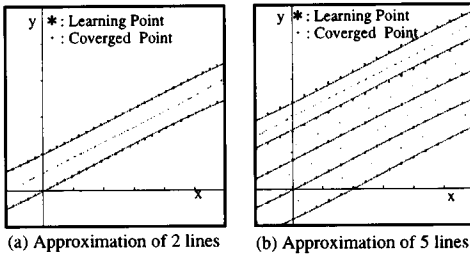


図6 任意価数の直線関数の近似
Fig. 6 Approximation of many valued linear functions.

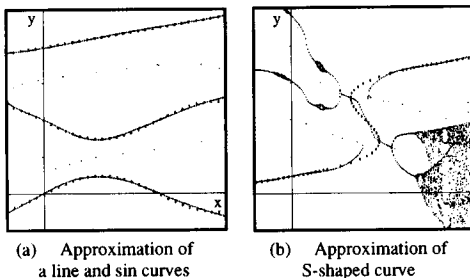


図7 多価関数の近似
Fig. 7 Approximation of many valued functions.

トワーク構造, 同じ学習条件で行った。

図6は, 前節のデータに対し直線の数を変えて, ネットワークが価数の変化に対応できるかを評価したものである。価数が増えるに従って, 誤差が十分に小さくなるまでの学習回数は増加するが, 隠れ層のニューロン数を固定したままで, 5本の直線(5価の関数)まで学習できることは確認できた。

図7には, データが曲線になる場合(a)と, 変数 x の値に応じて, 価数が1価, 3価, 1価と変化する S 次曲線の近似結果(b)を示した。両者とも, 学習は収束したが, S 次曲線では, 価数が少なくなる領域で状態が不安定になり, 近似曲線上に出力を引き込むことはできなかった。

5. シミュレーション結果の考察

シミュレーション結果から, 提案モデルが, 価数変化がない場合は, 任意の価数の多価関数を近似できることは確認できた。しかし, 価数に変化がある場合, 不安定な挙動を示す場合もあることがわかった。

図8は, 不安定な動作を示す領域($x=4$)での関数形 $y=f(4,z)$ のグラフを示したものである。図中, “*”で示されるのが学習データ, “●”で示されるのが変動データである。学習される関数 $f(x,z)$ に対し,

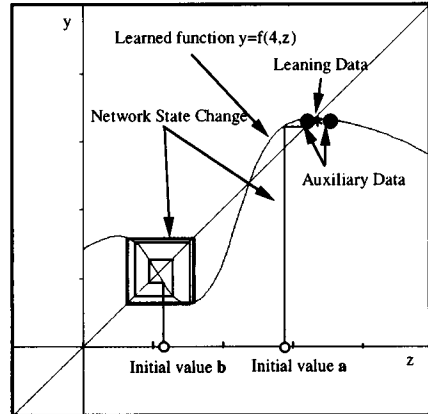


図8 不安定部分の動作解析
Fig. 8 Network dynamics in unstable region.

データは, 局所的にしか与えられない。データのある部分では, 漸近条件は満たされ, 広い平坦領域が形成されている。従って, データ近傍の初期値 a でネットワークを動作させると, 出力 y は, データ点に収束する。しかし, この例では, データのない領域でも, $y=z$ のグラフと交点が発生している。この場合, 初期値 b で動作させると, 交点の周りで状態が周回し, 収束は得られない。

価数変化のある多価関数の学習の難しさは, 志沢によっても指摘されている。志沢のモデルは, 価数を仮定したモデルなので, 価数変化への対応は原理的に不可能である。提案モデルは, 価数を仮定しないので原理的には学習可能であるが, この問題は, 引込み領域制御の問題に置き換えられる。現状, データのない領域での関数形成に関してはネットワークまかせになっており, これが, 不安定動作を引き起こす要因の一つになっているものと判断する。

しかし, 提案モデルは, 学習に漸近条件を含めることで, 引込み条件を学習することを可能にした。変動データの広がりや偏微分条件に対する重み β を調整することで, 引込み領域を制御することは可能である。学習条件によって, 引込み領域がどのように変化するか, 不安定現象の発生を, どのように抑制するかは今後の課題である。

6. むすび

本論文において, 多価関数を近似するフィードバック付き階層形NNモデルを提案した。このモデルは, 補助入力 z の初期値を変えることによって, 同じ入力

x に対して異なる出力 y を漸近的に得る。また、学習時には、出力教師データを補助入力として与え、補助入力に関する偏微分条件を荷重更新則に付加することによって、学習データがネットワークの安定点となるようにする。

シミュレーションにより、このモデルが、同一構造のモデルで、価数変化のない任意の多価関数を近似できること、また、補助入力の変動をデータを加えることで、広い収束領域が確保できることが確認できた。

しかし、価数変化があるときには、不安定な動作を引き起こす場合もある。今後、価数変化への対応、引き込み領域の制御に関して検討を行う予定である。

謝辞 本研究にあたり、御討議頂いた金沢大学自然科学研究科秋山等さんおよび金沢大学工学部中山研究室の皆さんに感謝します。

文 献

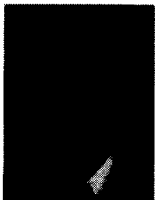
- [1] S. Haykin, Neural Networks, IEEE PRESS, 1994.
- [2] R.A. Jacobs and M.I. Jordan, "Learning piecewise control strategies in a modular neural network architecture," IEEE Trans. Sys. Man. and Cybern., vol.SMC-23, no.2, pp.337-345, 1993.
- [3] D. Lin, J.E. Dayhoff, and P.A. Ligomenides, "Trajectory production with the adaptive time-delay neural network," IEEE Trans. Sys. Man. and Cybern., vol.SMC-23, no.2, pp.337-345, 1993.
- [4] 志沢雅彦, "標準正則化理論の多価関数への拡張—滑らかな多重表面の復元," 信学論 (D-II), vol.J77-D-II, no.6, pp.1146-1156, June 1994.
- [5] 志沢雅彦, "多価正則化ネットワーク—多対 h 写像を学習する多層ネットワークの理論," 信学論 (A), vol.J78-A, no.3, pp.427-439, March 1995.
- [6] 富川義弘, 秋山 等, 中山謙二, "多価関数を実現するフィードバック付き階層形ニューラルネットワーク," 信学技報, NC95-167, March 1996.
- [7] B. Carnahan, H. A. Luther, and J. O. Wilkes, 藤田 宏訳, "計算機による数値計算法," 日本コンピュータ協会, 1982.

(平成 8 年 11 月 5 日受付)



中山 謙二 (正員)

昭 46 東工大・工・電子卒。昭 46~47 同大学研究生。昭 58 工博 (東工大)。昭 47 日本電気 (株) 入社。伝送通信事業部および C&C システム研究所に勤務し、通信用各種フィルターおよびディジタル信号処理の研究に従事。昭 63 金沢大・工・電気情報工学科助教授、平 2 教授となり現在に至る。最近の研究テーマは、適応信号処理およびニューラルネットワーク。昭 62 年 9 月 IEEE Circuits & Devices Mag. 論文賞受賞。著書「SC 回路網の設計と応用」(東海大学出版) ほか。IEEE シニア会員。



富川 義弘 (正員)

昭 57 名大・理・地球科学卒。同年、YKK (株) に入社。以来、同社研究開発本部において、信号処理、数値解析、製品検査の研究に従事。平 6 金沢大大学院・自然科学研究科・博士課程に入学。現在、ニューラルネットを利用した形状認識、および、その理論解析の研究に従事。